

Phylanx

Project meeting April 19th-20th, 2018

Hartmut Kaiser (hkaiser@cct.lsu.edu)

Phylanx: An Asynchronous Distributed Array Processing Toolkit

- High Performance Computing Challenges
 - Algorithms: need to be made work in distributed, requires data tiling
 - Programming Languages and Models: don't directly support distributed execution
 - Heterogeneous hardware: difficult to deal with various programming models
- Domain experts, specially in the field of machine learning, have traditionally shied away from utilizing HPC resources due to such challenges
- HPC resources are (becoming) the only viable solution with the ever increasing size of datasets.
- Goal: Abstract away complexities of programming on High Performance Computing resources from domain experts.

Phylanx: An Asynchronous Distributed Array Processing Toolkit

- Combine performance of HPC systems with the ease of programming in a high level language
- Python frontend to abstract away complexities of lower level implementations
 - Integration with Jupyter notebooks
- Run NumPy code directly in Phylanx
- Distributed task graphs are generated from Python
- HPX acts as the execution engine to execute the task graphs
- Promising initial results with execution time comparable to NumPy on shared memory systems.

An Asynchronous Distributed Array Processing Toolkit

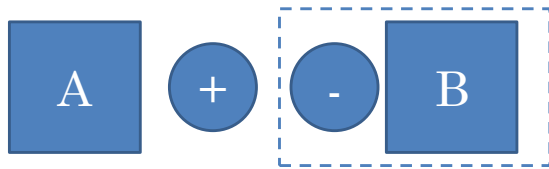
- Asynchronous
 - Interactive, jobs submitted to possibly remote resources
- Distributed
 - Utilize distributed compute resources: cloud and tightly coupled clusters
- Array Processing
 - Usable for arbitrary array processing tasks and algorithms, mainly machine learning
- Toolkit
 - Flexible, universal

Phylanx Structure

Frontend

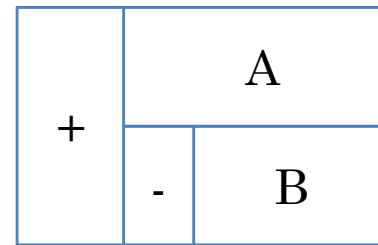
Expression: $A + (-B)$

Matrices A and B



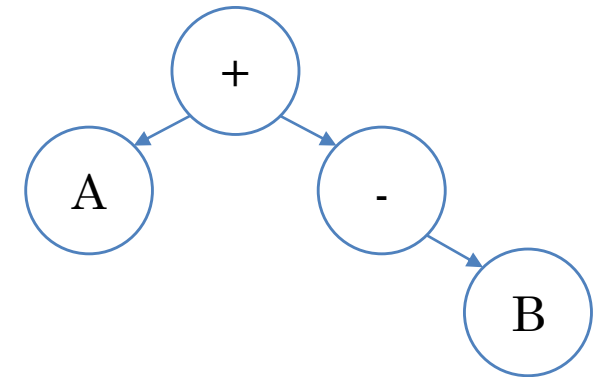
Middleware

Internal representation
(Abstract Syntax Tree)

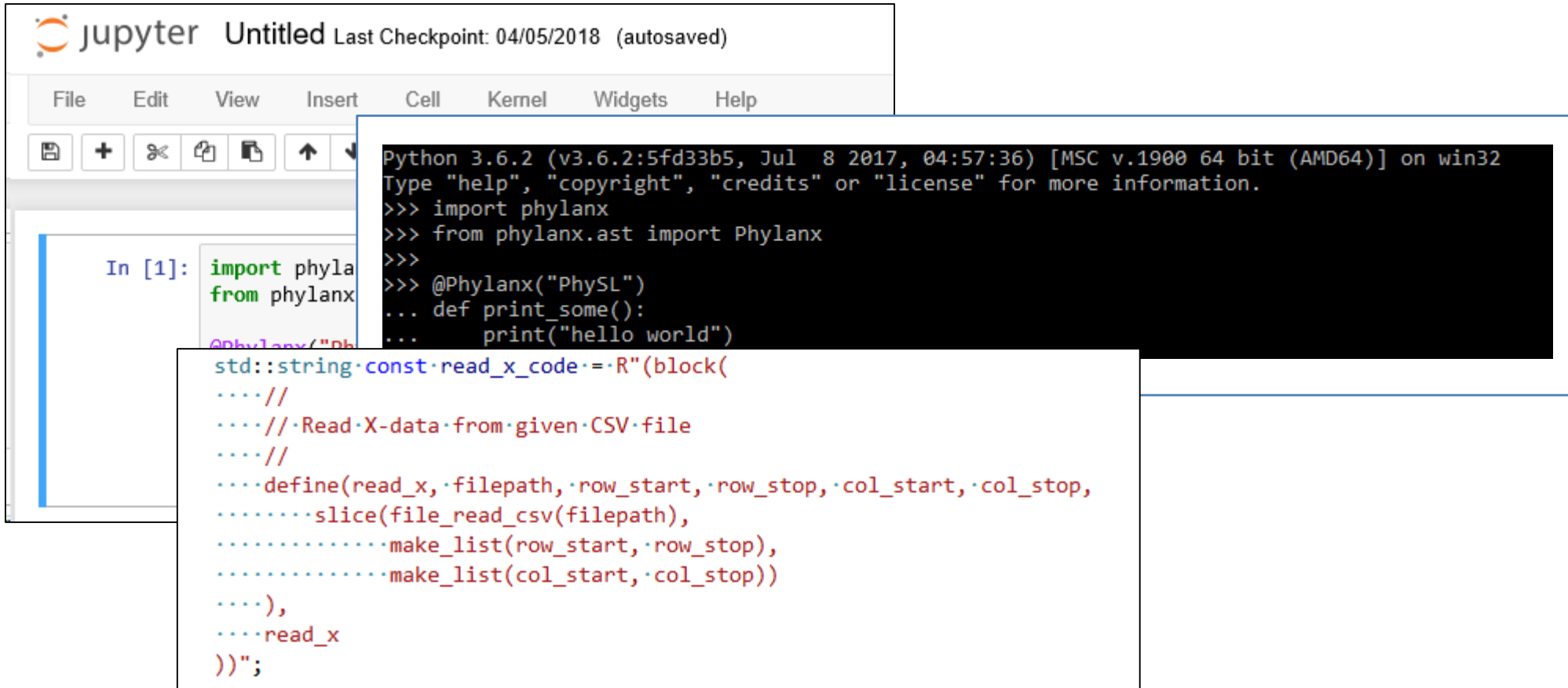


Backend

(Distributed) Execution Tree



Phylanx: Frontend



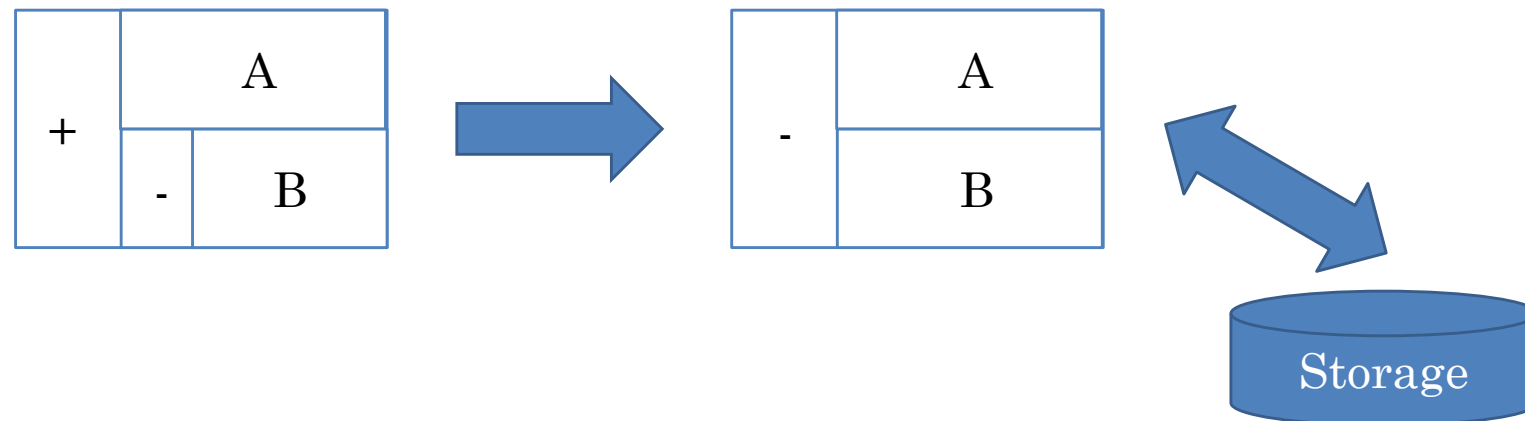
```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import phylanx
>>> from phylanx.ast import Phylanx
>>>
>>> @Phylanx("PhySL")
... def print_some():
...     print("hello world")
... 
```

```
In [1]: import phyla
        from phylanx
        @Phylanx("PhySL")
        def print_some():
            print("hello world")
        
```

```
std::string·const·read_x_code·=·R"(block(
····//
····//·Read·X-data·from·given·CSV·file
····//
····define(read_x,·filepath,·row_start,·row_stop,·col_start,·col_stop,
········slice(file_read_csv(filepath),
··········make_list(row_start,·row_stop),
··········make_list(col_start,·col_stop))
····),
····read_x
····);
```

Phylanx: Middleware

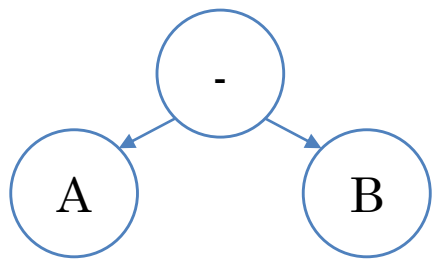
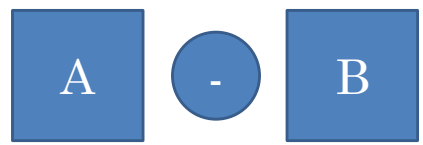
- Various transformations
 - Optimizations
 - Data Tiling and distribution
- Goal: Minimize computation and communication
- Specific for expression to be evaluated



Phylanx: Backend

- Adaptive, asynchronous execution using HPX
 - Maximum speed, pure C++

Single System



Distributed System (tiled data)

